文章编号: 1004-4353(2020)04-0344-08

基于蒙特卡洛模拟算法的欧冠淘汰赛抽签概率的研究

潘素娟[1,2]， 丁杰[3]
(1.福建商学院 信息工程学院，福建 福州 350012；2.金融数学福建省高校重点实验室(莆田学院)，福建 莆田 351100；3.厦门大学 经济学院，福建 厦门 361005)

摘要: 为研究欧冠淘汰赛抽签概率问题,以欧冠2017—2018赛季淘汰赛的抽签规则和抽签流程为例,利用蒙特卡洛模拟算法建立了一种新型的抽签概率模型,并通过计算得出了对阵概率的数值解.利用置信区间和分位点对模型所得的对阵概率进行验证表明,该模拟方法具有较好的可信度.对抽签概率进行模拟显示,巴萨队对阵切尔西队的概率约为40%(该结果与实际对阵结果相符),由此再次表明该模拟方法具有可靠性.

关键词: 抽签概率；蒙特卡洛模拟；区间估计；置信区间；分位点
中图分类号: O212.2　　**文献标识码:** A

Research on the draw probability of Champions League knockout matches based on Monte Carlo simulation algorithm

PAN Sujuan[1,2]， DING Jie[3]
(1. *Department of Information Engineering, Fujian Commercial College, Fuzhou* 350012, *China*; 2. *Key Laboratory of Financial Mathematics of Fujian Province University (Putian University)*, *Putian* 351100, *China*; 3. *School of Economics, Xiamen University, Xiamen* 361005, *China*)

Abstract: In order to investigate the draw probability of the Champions League knockout tournament, a new draw probability model is established by using Monte Carlo simulation algorithm, and it is used in the draw rules and process of the 2017-2018 Champions League knockout round, thus the numerical solution of the match probability is obtained through calculation. The confidence interval and quantile are used to verify the match probability of the model, the proposed method exhibit good reliability. The simulation of the draw probability of Barcelona versus Chelsea is about 40%, which is consistent with the final actual result, therefroe the simulation results once more indicate that the proposed method is reliable.

Keywords: draw probability; Monte Carlo simulation; interval estimation; confidence interval; quantile

0　引言

研究抽签概率不仅能促进统计和概率理论的发展,同时也能解决许多现实的问题,比如招标或投标项目、解决城市有限公共资源的供需矛盾、构建和运行马拉松赛事信息系统、设计和优化政府征收等.目前,国内外已有许多学者对抽签概率进行了研究,并取得了较好的研究结果[1-6].在体育竞技淘汰赛中,竞赛对象的不同可能会直接影响到竞赛的结果.为了解决竞赛过程中出现的机遇性强和竞赛结果的偶然性大等问题,通常采用抽

收稿日期: 2020-06-20　**作者简介:** 潘素娟(1982—),女,副教授,研究方向为概率统计、随机过程与金融数学.
基金项目: 福建省中青年教师教育科研项目(JAT190502);金融数学福建省高校重点实验室开放课题立项项目(JR201802);国家自然科学基金资助项目(11001142)

签来决定竞赛对象,以此最大限度地保证竞赛过程的公正性和竞赛结果的合理性. 目前,针对体育竞技淘汰赛的抽签方法主要有“1/4 区公式控制抽签法”“1/2 区逐级分区抽签法”“逐区双分抽签法”等方法. 其中:“1/4 区公式控制抽签法”能够较好地解决淘汰制抽签中“机遇”与“控制”的矛盾,但该方法的抽签过程较为繁琐,且进区概率存在失真性等问题[7].“1/2 区逐级分区抽签法”虽然能够克服“1/4 区公式控制抽签法”所存在的缺陷,但该方法并未考虑受随机因素影响的淘汰赛抽签[8].“逐区双分抽签法”虽然能够使得整个抽签过程做到最大限度的随机,但该方法并未考虑很多淘汰赛规则中所隐含的陷阱信息[9]. 目前为止,对淘汰赛中所隐含的陷阱还没有较好的模拟方法可以实现其对阵概率的求解.

蒙特卡洛模拟算法[10]是以概率统计理论为基础的一种计算方法,它可以随机模拟各种变量间的动态关系,把每一种不确定性对结果的影响以概率分布的形式表示出来,因此该方法可以解决数值解的求解问题. 目前,已有学者利用蒙特卡洛模拟方法研究了彩票的概率模型,并取得了较好的效果[11],但利用该方法来研究抽签概率的研究尚未见到报道. 因此,基于上述难以求出对阵概率解析解的抽签概率问题,本文以 2017—2018 赛季的欧冠淘汰赛抽签概率为例,利用蒙特卡洛模拟方法求出进入欧冠淘汰赛的各支球队之间的对阵概率,并分别利用置信区间和分位点对模拟结果的可信度进行了分析.

1　研究设计

1.1　研究对象

欧冠 2017—2018 赛季的淘汰赛一共有 16 支球队. 小组赛中以小组第 1 身份出线的球队有:曼联(A 组,英超)、巴黎(B 组,法甲)、罗马(C 组,意甲)、巴萨(D 组,西甲)、利物浦(E 组,英超)、曼城(F 组,英超)、贝西克塔斯(G 组,土超)和热刺(H 组,英超);小组赛中以小组第 2 身份出线的球队有:巴塞尔(A 组,瑞超)、拜仁(B 组,德甲)、切尔西(C 组,英超)、尤文(D 组,意甲)、塞维利亚(E 组,西甲)、矿工(F 组,乌超)、波尔图(G 组,葡超)和皇马(H 组,西甲).

1.2　抽签规则

欧冠 2017—2018 赛季的淘汰赛采用抽签的办法确定竞赛对象. 若对所有球队都不做任何约束条件,则每一支球队都可能抽到任何一个竞争对象,即抽签只是为了保证所有球队都有一个同等的机遇条件. 但比赛组织方对欧冠 2017—2018 赛季淘汰赛的抽签制定了如下基本规则:

1)小组第 1 的球队对阵小组第 2 的球队.

2)同个小组的球队之间回避,如曼联不能和小组赛同组的巴塞尔对阵.

3)同国联赛的球队规避,如巴萨不能与同为西甲的皇马对决.

1.3　模型的建立

建立模型的目标是计算一个 8×8 的矩阵. 矩阵内的元素对应的是两支球队之间的对阵概率,矩阵的行对应的是小组第 1 出线的球队(曼联标记为 1,巴黎标记为 2,罗马标记为 3,巴萨标记为 4,利物浦标记为 5,曼城标记为 6,贝西克塔斯标记为 7,热刺标记为 8),矩阵的列对应的是小组第 2 出线的球队(巴塞尔标记为 1,拜仁标记为 2,切尔西标记为 3,尤文标记为 4,塞维利亚标记为 5,矿工标记为 6,波尔图标记为 7,皇马标记为 8).

1.4　抽签流程

抽签的流程为:第 1 轮,先从小组第 1 的球队中抽出 1 支球队,然后从它可能对应的(要考虑到规则 2 和规则 3 中的回避条款)小组第 2 中的对手中抽出 1 支球队,然后这两支球队组成对阵双方;第 2 轮,在剩下的球队中再次进行上一步操作. 依据上述流程,只有完成第 8 轮抽签才可完成对阵抽签,但在实际抽签过程中会存在如下陷阱:

1)虽然理论上要进行 8 轮抽签才能确定最后的对阵,但在实际中抽到第 7 轮就已经能够确定最后的对阵情况. 其原因是,7 轮抽签过后小组第 1 和小组第 2 的球队都只剩下 1 支,所以肯定是他们之间进行对决.

2)假设 6 轮过后,在剩下的球队中小组第 1 的球队是巴萨和贝西克塔斯,小组第 2 的球队是矿工和皇马. 在这种情况下,经过 6 轮抽签即可确定对阵情况. 因为小组第 1 的巴萨队不能对阵皇马队(同国联赛回避),所以巴萨队只能对阵矿工队,由此可以推出贝西克塔斯队对阵皇马队.

3)在陷阱 2)中,小组第 2 的皇马队不能对阵巴萨队(同国联赛回避),所以皇马队只能对阵贝西克塔斯队,由此可以推出巴萨队对阵矿工队.

陷阱 2)和 3)在规则方面较为接近,但在有些情形中两者并不一致.例如:假设前 2 轮抽签结束后,对阵结果是巴黎队对阵皇马队,贝西克塔斯队对阵拜仁队.在这种情况下,由于存在同组规避和同国联赛规避的条款,因此小组第 2 的切尔西队对阵的球队只有巴黎、巴萨和贝西克塔斯,然而在巴黎队和贝西克塔斯队都已经被抽走的情况下,与切尔西队对阵的只能是巴萨队.上述情形所对应的就是陷阱 3),而非陷阱 2).根据欧冠规则,同一个联赛原则上最多的欧冠名额是“3+1”个,其中 3 个球队直接进入欧冠小组赛,另 1 个球队参加附加赛(如果附加赛获胜就可以参加小组赛).另外,如果该联赛有 1 支其他的球队在上一年获得了欧联杯的冠军,那么该支球队也可以直接获得参加欧冠小组赛的资格.所以从理论上说,某个联赛参加欧冠小组赛的球队最多可以是 5 支.如果这 5 支球队都能小组出线,那么该联赛有 5 支球队能够参加欧冠淘汰赛.如果其中 4 支球队为小组第 1,另 1 支球队为小组第 2;或者其中 4 支球队为小组第 2,另 1 支球队小组第 1:那么和其他 4 支球队不在同一个组的那支球队不仅需要回避 4 支同国联赛的球队,还要回避同小组的球队.所以该球队需要回避的球队有 5 支,能够选择对阵的球队只有 3 支.这种最极端的例子出现在欧冠 2017—2018 赛季的淘汰赛抽签中,即切尔西队就是那支需要回避 5 支球队的球队.在这种最极端的情形下,经过两轮抽签后就可决定球队的对阵情况,即至少经过 2 轮抽签过后才可能会出现陷阱 2)或陷阱 3)的现象.

2 模拟过程

2.1 利用蒙特卡洛模拟计算对阵概率数值解的步骤

第 1 步　生成 1 个 8×8 且元素全部为 0 的矩阵.

第 2 步　根据规则先抽出一组对阵结果(即 16 支球队的 8 场对阵结果).如果 2 支球队之间的抽签结果为对阵,则在矩阵中对与该结果相对应的元素加 1;如果 2 支球队之间抽签的结果不为对阵,则矩阵中相应的元素不变.

第 3 步　重复步骤 2,直到完成 100 000 次蒙特卡罗模拟.

第 4 步　将步骤 3 最终得到的矩阵中每个元素的最后累积值除以蒙特卡洛模拟的次数(100 000次),即可得到对阵概率的数值解.

2.2 编程

2.1 模拟步骤中的第 2 步的编程方法为:①根据规则任意抽取前两轮(因在最极端的情形下,前两轮也不会掉进陷阱 2)或者陷阱 3)).②从第 3 轮开始,首先观察每个小组第 1 的球队会不会出现只剩下 1 个可以选择的对手的情形.如果有,直接抽取出来;如果没有,就进入下一步.然后观察每个小组第 2 的球队会不会出现只剩下一个可以选择的对手的情形.如果有,直接抽取出来;如果没有,就可以和前两轮一样根据规则(即在同组回避和同国联赛规避的条件下)进行抽取.③第 4 轮到第 7 轮的抽签思路与第 3 轮相同.在所有的情形下,实际上不需要进行第 8 轮抽签,因为经过第 7 轮抽签后就可确定最终的对阵结果.经编程(见附录)得到的对阵概率如表 1 所示.

表 1　各球队之间的对阵概率

	巴塞尔	拜仁	切尔西	尤文	塞维利亚	矿工	波尔图	皇马
曼联	0	0.145 63	0	0.184 35	0.185 10	0.154 85	0.146 34	0.183 73
巴黎	0.107 89	0	0.300 36	0.124 44	0.128 10	0.107 60	0.103 49	0.128 12
罗马	0.159 85	0.149 65	0	0	0.189 93	0.159 88	0.151 99	0.188 70
巴萨	0.150 69	0.149 52	0.402 19	0	0	0.151 53	0.146 07	0
利物浦	0.158 69	0.148 16	0	0.192 82	0	0.159 20	0.153 93	0.187 20
曼城	0.154 36	0.148 87	0	0.183 58	0.181 22	0	0.146 10	0.185 87
贝西克塔斯	0.109 49	0.106 71	0.297 45	0.125 08	0.125 16	0.109 73	0	0.126 38
热刺	0.159 03	0.151 46	0	0.189 73	0.190 49	0.157 21	0.152 08	0

3　可信度分析

利用蒙特卡罗模拟方法研究数值解问题时，若输入模式中的随机数并不是真正的随机数，则模拟及预测结果就会产生错误. 为此，本文利用区间估计的方法（置信区间和分位点）对上述模拟得到的对阵概率结果进行可信度分析.

1）置信区间法. 利用模型对对阵相关的 64 个参数进行 100 000 次蒙特卡洛模拟后发现，每次模拟都服从 I.I.D.假设[12]. 为了验证可信度，对上述对阵概率的置信区间进行 100 次实验，每次实验均进行 100 000 次蒙特卡洛模拟. 根据实验结果计算出的对阵概率的 95％水平下的置信区间和 99％水平下的置信区间的结果如表 2 和表 3 所示. 由表 2 和表 3 可以看出，表中的置信区间都比较狭窄，说明本文提出的模型具有较高的可信度.

表 2　对阵概率在 95％水平下的置信区间

	1	2	3	4
1	(0,0)	(14.687 5,14.729 8)	(0,0)	(18.334 6,18.387 7)
2	(10.749 1,10.788 6)	(0,0)	(29.916 2,29.971 2)	(12.646 9,12.687 9)
3	(15.870 0,15.915 4)	(15.094 5,15.137 3)	(0,0)	(0,0)
4	(15.200 9,15.241 1)	(14.695 7,14.741 9)	(40.076 4,40.137 6)	(0,0)
5	(15.884 7,15.933 5)	(15.092 3,15.132 7)	(0,0)	(18.974 6,19.019 1)
6	(15.467 1,15.513 6)	(14.696 1,14.741 5)	(0,0)	(18.343 0,18.388 3)
7	(10.788 3,10.825 7)	(10.473 5,10.512 4)	(29.920 7,29.977 9)	(12.622 2,12.661 2)
8	(15.887 9,15.934 0)	(15.108 5,15.156 3)	(0,0)	(18.942 4,18.992 0)

	5	6	7	8
1	(18.332 3,18.385 3)	(15.445 7,15.497 0)	(14.695 7,14.745 2)	(18.357 5,18.401 8)
2	(12.651 9,12.692 4)	(10.788 0,10.829 6)	(10.460 7,10.497 5)	(12.639 9,12.679 9)
3	(18.955 8,19.002 2)	(15.886 7,15.929 0)	(15.113 8,15.156 8)	(18.943 5,18.995 1)
4	(0,0)	(15.201 5,15.246 8)	(14.707 7,14.750 3)	(0,0)
5	(0,0)	(15.876 2,15.921 5)	(15.099 5,15.144 3)	(18.935 8,18.985 8)
6	(18.318 4,18.366 8)	(0,0)	(14.685 8,14.728 9)	(18.352 6,18.397 9)
7	(12.651 5,12.692 4)	(10.758 6,10.805 5)	(0,0)	(12.632 8,12.677 5)
8	(18.952 0,18.999 0)	(15.884 5,15.929 5)	(15.084 0,15.129 9)	(0,0)

表 3　对阵概率在 99％水平下的置信区间

	1	2	3	4
1	(0,0)	(14.680 9,14.736 4)	(0,0)	(18.326 3,18.396 0)
2	(10.742 9,10.794 8)	(0,0)	(29.907 6,29.979 9)	(12.640 5,12.694 4)
3	(15.862 9,15.922 5)	(15.087 8,15.144 0)	(0,0)	(0,0)
4	(15.194 6,15.247 5)	(14.688 4,14.749 2)	(40.066 8,40.147 2)	(0,0)
5	(15.877 0,15.941 2)	(15.085 9,15.139 1)	(0,0)	(18.9676,19.026 1)
6	(15.459 8,15.521 0)	(14.688 9,14.748 6)	(0,0)	(18.335 8,18.395 5)
7	(10.782 4,10.831 6)	(10.467 4,10.518 5)	(29.911 7,29.986 8)	(12.616 1,12.667 3)
8	(15.880 6,15.941 3)	(15.101 0,15.163 8)	(0,0)	(18.934 6,18.999 8)

	5	6	7	8
1	(18.324 0,18.393 6)	(15.437 6,15.505 1)	(14.687 9,14.752 9)	(18.350 5,18.408 8)
2	(12.645 5,12.698 8)	(10.781 4,10.836 2)	(10.454 9,10.503 3)	(12.633 7,12.686 2)
3	(18.948 5,19.009 5)	(15.880 1,15.935 6)	(15.107 0,15.163 5)	(18.935 4,19.003 2)
4	(0,0)	(15.194 4,15.253 9)	(14.701 0,14.757 0)	(0,0)
5	(0,0)	(15.869 1,15.928 6)	(15.092 5,15.151 3)	(18.927 9,18.993 7)
6	(18.310 9,18.374 4)	(0,0)	(14.679 0,14.735 7)	(18.345 5,18.405 1)
7	(12.645 0,12.698 9)	(10.751 2,10.812 8)	(0,0)	(12.625 8,12.684 5)
8	(18.944 6,19.006 4)	(15.877 4,15.936 6)	(15.076 8,15.137 1)	(0,0)

2)分位点法. 以下通过分析两组分位点的结果来验证模拟结果的可信度. 第 1 组为 2.5 百分位点到 97.5 百分位点,如表 4 所示;第 2 组为 0.5 百分位点到 99.5 百分位点,如表 5 所示. 由表 4 和表 5 可以看出,表中的区间都比较狭窄,该结果再次说明本文提出的模拟方法具有较高的可信度.

表 4 对阵概率在 2.5 到 97.5 的百分位点

	1	2	3	4
1	(0,0)	(14.526 4,14.901 0)	(0,0)	(18.106 3,18.587 2)
2	(10.569 8,10.966 6)	(0,0)	(29.650 0,30.221 9)	(12.460 2,12.858 7)
3	(15.654 7,16.098 9)	(14.943 2,15.354 6)	(0,0)	(0,0)
4	(15.001 6,15.407 4)	(14.508 4,14.932 3)	(39.823 7,40.386 5)	(0,0)
5	(15.660 7,16.144 2)	(14.872 9,15.300 2)	(0,0)	(18.820 9,19.255 5)
6	(15.245 7,15.696 7)	(14.500 2,14.902 6)	(0,0)	(18.134 4,18.581 2)
7	(10.638 7,10.995 0)	(10.259 3,10.691 5)	(29.655 5,30.198 7)	(12.436 1,12.845 2)
8	(15.667 7,16.137 6)	(14.920 9,15.328 4)	(0,0)	(18.758 6,19.202 2)

	5	6	7	8
1	(18.138 5,18.619 7)	(15.222 9,15.694 8)	(14.467 1,14.950 1)	(18.194 8,18.585 6)
2	(12.463 7,12.846 2)	(10.622 9,10.998 5)	(10.282 6,10.655 6)	(12.479 3,12.881 8)
3	(18.763 8,19.208 1)	(15.684 3,16.105 4)	(14.920 5,15.323 0)	(18.728 6,19.208 9)
4	(0,0)	(15.020 3,15.459 6)	(14.518 2,14.955 7)	(0,0)
5	(0,0)	(15.673 8,16.107 6)	(14.880 6,15.325 0)	(18.738 3,19.198 0)
6	(18.087 5,18.579 7)	(0,0)	(14.518 4,14.892 2)	(18.154 2,18.578 1)
7	(12.505 6,12.908 6)	(10.564 7,11.026 7)	(0,0)	(12.438 9,12.870 1)
8	(18.751 9,19.226 6)	(15.688 2,16.096 3)	(14.890 9,15.334 8)	(0,0)

表 5 对阵概率在 0.5 到 99.5 的百分位点

	1	2	3	4
1	(0,0)	(14.512 0,14.926 6)	(0,0)	(17.981 4,18.649 5)
2	(10.525 0,10.998 6)	(0,0)	(29.582 3,30.306 2)	(12.443 5,12.921 1)
3	(15.629 5,16.224 6)	(14.895 4,15.415 4)	(0,0)	(0,0)
4	(14.973 8,15.441 1)	(14.473 0,15.004 0)	(39.753 3,40.409 0)	(0,0)
5	(15.613 9,16.166 6)	(14.827 4,15.316 6)	(0,0)	(18.741 3,19.292 1)
6	(15.172 0,15.758 6)	(14.439 1,14.917 1)	(0,0)	(18.096 3,18.678 8)
7	(10.568 4,11.040 1)	(10.216 2,10.728 1)	(29.614 2,30.257 5)	(12.405 5,12.884 5)
8	(15.618 4,16.168 1)	(14.788 8,15.392 2)	(0,0)	(18.667 6,19.238 6)

	5	6	7	8
1	(18.066 9,18.647 1)	(15.151 4,15.736 1)	(14.419 1,15.026 2)	(18.111 2,18.654 9)
2	(12.412 6,12.933 4)	(10.571 8,11.057 1)	(10.188 8,10.682 1)	(12.440 7,12.938 4)
3	(18.731 9,19.251 6)	(15.660 0,16.185 1)	(14.894 4,15.345 1)	(18.650 0,19.267 2)
4	(0,0)	(15.001 0,15.563 2)	(14.483 4,15.051 7)	(0,0)
5	(0,0)	(15.627 0,16.137 8)	(14.824 2,15.385 1)	(18.701 8,19.288 7)
6	(18.083 5,18.631 7)	(0,0)	(14.492 4,14.936 3)	(18.077 9,18.586 0)
7	(12.427 1,12.942 1)	(10.518 8,11.105 5)	(0,0)	(12.427 5,12.908 1)
8	(18.711 8,19.275 6)	(15.620 7,16.169 7)	(14.852 7,15.398 5)	(0,0)

4 结论

本文通过对欧冠 2017—2018 赛季淘汰赛的抽签规则和抽签流程进行分析,利用蒙特卡洛模拟算法建立了一种新型的抽签概率模型,并给出了对阵概率的数值解. 利用置信区间和分位点对模型计算所得的对阵概率进行可信度分析表明,本文的模拟方法具有较好的可信度. 模拟结果显示,巴萨与切尔西对阵的概率约为 40%(该结果与实际比赛结果相符),由此再次表明本文模拟方

法具有可靠性.另外，由于本研究已经考虑了最极端的情形，所以对于其他欧冠赛季，在不改比赛规则的情况下只要改变初始数据集就可以得到对阵概率.本文方法也可以为金融市场的风险管理、项目的招标与投标等问题的抽签提供借鉴.

附录：

R 代码：

```
library(lubridate)
set.seed(1234)
sink(file="C:/Users/Ding/Desktop/UEFA Champions League.txt",append=T)
n_1 <- 100000 # 100000 times Monte Carlos Simulations each experiment
n_2 <- 100 # 100 experiments
now()
result_all <- array(NA,dim=c(8,8,n_2))
for (x in 1:n_2) {
  result_cum <- matrix(0,nrow=8,ncol=8)
    FUN_1 <- function(a,b) {
    result[a,1] <<- set_1[b]
    result[a,2] <<- set_2_2[[set_1[b]]]
    set_1 <<- setdiff(set_1,result[a,1])
    for (j in 1:8) {
      set_1_1[[j]] <<- setdiff(set_1_1[[j]],result[a,1])
    }
    set_2 <<- setdiff(set_2,result[a,2])
    for (j in 1:8) {
      set_2_2[[j]] <<- setdiff(set_2_2[[j]],result[a,2])
    }
  }
    FUN_2 <- function(a,b) {
    result[a,1] <<- set_1_1[[set_2[b]]]
    result[a,2] <<- set_2[b]
    set_1 <<- setdiff(set_1,result[a,1])
    for (j in 1:8) {
      set_1_1[[j]] <<- setdiff(set_1_1[[j]],result[a,1])
    }
    set_2 <<- setdiff(set_2,result[a,2])
    for (j in 1:8) {
      set_2_2[[j]] <<- setdiff(set_2_2[[j]],result[a,2])
    }
  }
    for (k in 1:n_1) {
    result <- matrix(NA,nrow=8,ncol=2)
    set_1 <- 1:8
    set_2 <- 1:8
    set_1_1 <- list(setdiff(1:8,1),setdiff(1:8,2),setdiff(1:8,c(1,3,5,6,8)),setdiff(1:8,c(3,4)),setdiff(1:8,c(4,5)),setdiff(1:8,6),setdiff(1:8,7),setdiff(1:8,c(4,8)))
    set_2_2 <- list(setdiff(1:8,c(1,3)),setdiff(1:8,2),setdiff(1:8,c(3,4)),setdiff(1:8,c(4,5,8)),setdiff(1:8,c(3,5)),setdiff(1:8,c(3,6)),setdiff(1:8,7),setdiff(1:8,c(3,8)))
    for (i in 1:2) {
      result[i,1] <- sample(set_1,1)
      result[i,2] <- sample(set_2_2[[result[i,1]]],1)
      set_1 <- setdiff(set_1,result[i,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[i,1])
      }
      set_2 <- setdiff(set_2,result[i,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[i,2])
      }
    }
    if (length(set_2_2[[set_1[1]]])==1) {
      FUN_1(3,1)
    } else if (length(set_2_2[[set_1[2]]])==1) {
      FUN_1(3,2)
    } else if (length(set_2_2[[set_1[3]]])==1) {
      FUN_1(3,3)
    } else if (length(set_2_2[[set_1[4]]])==1) {
      FUN_1(3,4)
    } else if (length(set_2_2[[set_1[5]]])==1) {
      FUN_1(3,5)
    } else if (length(set_2_2[[set_1[6]]])==1) {
      FUN_1(3,6)
    } else if (length(set_1_1[[set_2[1]]])==1) {
      FUN_2(3,1)
    } else if (length(set_1_1[[set_2[2]]])==1) {
      FUN_2(3,2)
    } else if (length(set_1_1[[set_2[3]]])==1) {
      FUN_2(3,3)
    } else if (length(set_1_1[[set_2[4]]])==1) {
      FUN_2(3,4)
    } else if (length(set_1_1[[set_2[5]]])==1) {
      FUN_2(3,5)
    } else if (length(set_1_1[[set_2[6]]])==1) {
      FUN_2(3,6)
    } else {
      result[3,1] <- sample(set_1,1)
      result[3,2] <- sample(set_2_2[[result[3,1]]],1)
      set_1 <- setdiff(set_1,result[3,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[3,1])
      }
      set_2 <- setdiff(set_2,result[3,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[3,2])
      }
    }
    if (length(set_2_2[[set_1[1]]])==1) {
      FUN_1(4,1)
    } else if (length(set_2_2[[set_1[2]]])==1) {
      FUN_1(4,2)
    } else if (length(set_2_2[[set_1[3]]])==1) {
      FUN_1(4,3)
    } else if (length(set_2_2[[set_1[4]]])==1) {
      FUN_1(4,4)
    } else if (length(set_2_2[[set_1[5]]])==1) {
      FUN_1(4,5)
    } else if (length(set_1_1[[set_2[1]]])==1) {
      FUN_2(4,1)
    } else if (length(set_1_1[[set_2[2]]])==1) {
      FUN_2(4,2)
    } else if (length(set_1_1[[set_2[3]]])==1) {
```

```
      FUN_2(4,3)
    } else if (length(set_1_1[[set_2[4]]])==1) {
      FUN_2(4,4)
    } else if (length(set_1_1[[set_2[5]]])==1) {
      FUN_2(4,5)
    } else {
      result[4,1] <- sample(set_1,1)
      result[4,2] <- sample(set_2_2[[result[4,1]]],1)
      set_1 <- setdiff(set_1,result[4,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[4,1])
      }
      set_2 <- setdiff(set_2,result[4,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[4,2])
      }
    }
    if (length(set_2_2[[set_1[1]]])==1) {
      FUN_1(5,1)
    } else if (length(set_2_2[[set_1[2]]])==1) {
      FUN_1(5,2)
    } else if (length(set_2_2[[set_1[3]]])==1) {
      FUN_1(5,3)
    } else if (length(set_2_2[[set_1[4]]])==1) {
      FUN_1(5,4)
    } else if (length(set_1_1[[set_2[1]]])==1) {
      FUN_2(5,1)
    } else if (length(set_1_1[[set_2[2]]])==1) {
      FUN_2(5,2)
    } else if (length(set_1_1[[set_2[3]]])==1) {
      FUN_2(5,3)
    } else if (length(set_1_1[[set_2[4]]])==1) {
      FUN_2(5,4)
    } else {
      result[5,1] <- sample(set_1,1)
      result[5,2] <- sample(set_2_2[[result[5,1]]],1)
      set_1 <- setdiff(set_1,result[5,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[5,1])
      }
      set_2 <- setdiff(set_2,result[5,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[5,2])
      }
    }
    if (length(set_2_2[[set_1[1]]])==1) {
      FUN_1(6,1)
    } else if (length(set_2_2[[set_1[2]]])==1) {
      FUN_1(6,2)
    } else if (length(set_2_2[[set_1[3]]])==1) {
      FUN_1(6,3)
    } else if (length(set_1_1[[set_2[1]]])==1) {
      FUN_2(6,1)
    } else if (length(set_1_1[[set_2[2]]])==1) {
      FUN_2(6,2)
    } else if (length(set_1_1[[set_2[3]]])==1) {
      FUN_2(6,3)
    } else {
      result[6,1] <- sample(set_1,1)
      result[6,2] <- sample(set_2_2[[result[6,1]]],1)
      set_1 <- setdiff(set_1,result[6,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[6,1])
      }
      set_2 <- setdiff(set_2,result[6,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[6,2])
      }
    }
    if (length(set_2_2[[set_1[1]]])==1) {
      FUN_1(7,1)
    } else if (length(set_2_2[[set_1[2]]])==1) {
      FUN_1(7,2)
    } else if (length(set_2_2[[set_1[3]]])==1) {
      FUN_1(7,3)
    } else if (length(set_1_1[[set_2[1]]])==1) {
      FUN_2(7,1)
    } else if (length(set_1_1[[set_2[2]]])==1) {
      FUN_2(7,2)
    } else if (length(set_1_1[[set_2[3]]])==1) {
      FUN_2(7,3)
    } else {
      result[7,1] <- sample(set_1,1)
      result[7,2] <- sample(set_2_2[[result[7,1]]],1)
      set_1 <- setdiff(set_1,result[7,1])
      for (j in 1:8) {
        set_1_1[[j]] <- setdiff(set_1_1[[j]],result[7,1])
      }
      set_2 <- setdiff(set_2,result[7,2])
      for (j in 1:8) {
        set_2_2[[j]] <- setdiff(set_2_2[[j]],result[7,2])
      }
    }
    result[8,1] <- setdiff(1:8,result[1:7,1])
    result[8,2] <- setdiff(1:8,result[1:7,2])
    for (i in 1:8) {
      result_cum[result[i,1],result[i,2]] <- result_cum[re-
sult[i,1],result[i,2]]+1
    }
    #print(k)
  }
  #print(x)
  print(now())
  result_all[,,x] <- result_cum/n_1
}
result_mean <- matrix(NA,nrow=8,ncol=8)
for (i in 1:8) {
  for (j in 1:8) {
    result_mean[i,j] <- mean(result_all[i,j,])
  }
}
result_sd <- matrix(NA,nrow=8,ncol=8)
for (i in 1:8) {
  for (j in 1:8) {
    result_sd[i,j] <- sd(result_all[i,j,])
  }
}
result_all[,,1]
#### Confidence Interval
ci_95 <- data.frame(cbind(rep(NA,8),rep(NA,8),rep(NA,
```

```
8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,
8)))
#ci denotes confidence interval
colnames(ci_95) <- as.character(1:8)
# 95% confidence interval
for (i in 1:8) {
  for (j in 1:8) {
    ci_95[i,j] <- paste0("(",round((result_mean+qnorm
(.025)*result_sd/sqrt(n_2))[i,j]*100,4),",",round((result_
mean+qnorm(.975)*result_sd/sqrt(n_2))[i,j]*100,4),")")
  }
}
ci_95 # in percentage
ci_99 <- data.frame(cbind(rep(NA,8),rep(NA,8),rep(NA,
8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,
8)))
colnames(ci_99) <- as.character(1:8)
# 99% confidence interval
for (i in 1:8) {
  for (j in 1:8) {
    ci_99[i,j] <- paste0("(",round((result_mean+qnorm
(.005)*result_sd/sqrt(n_2))[i,j]*100,4),",",round((result_
mean+qnorm(.995)*result_sd/sqrt(n_2))[i,j]*100,4),")")
  }
}
ci_99 # in percentage
#### Quantile
# From .025th quantile to .975th quantile
quantile_1 <- data.frame(cbind(rep(NA,8),rep(NA,8),rep
(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep
(NA,8)))
colnames(quantile_1) <- as.character(1:8)
for (i in 1:8) {
  for (j in 1:8) {
    quantile_1[i,j] <- paste0("(",round(quantile(result_all
[i,j,],.025)*100,4),",",round(quantile(result_all[i,j,],
.975)*100,4),")")
  }
}
quantile_1 # in percentage
#### From .005th quantile to .995th quantile
quantile_2 <- data.frame(cbind(rep(NA,8),rep(NA,8),rep
(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep(NA,8),rep
(NA,8)))
colnames(quantile_2) <- as.character(1:8)
for (i in 1:8) {
  for (j in 1:8) {
    quantile_2[i,j] <- paste0("(",round(quantile(result_all
[i,j,],.005)*100,4),",",round(quantile(result_all[i,j,],
.995)*100,4),")")
  }
}
quantile_2 # in percentage
sink()
rm(list=ls())
1/4+1/5.5+1/6+1/8.5+1/9+1/15+1/15+1/17+1/28+1/
34
```

注：

1）set_1 表示在第 1 小组中可抽的球队，set_2 表示在第 2 小组中可抽的球队.

2）set_1_1 表示一个长度为 8 的列表，其中的每一个元素都是一个向量. 元素 i 表示第 2 小组中的第 i 支球队目前能够对阵第 1 小组的球队编号.

3）set_2_2 表示一个长度为 8 的列表，其中的每一个元素都是一个向量. 元素 i 表示第 1 小组中的第 i 支球队目前能够对阵第 2 小组的球队编号.

4）为了提高代码的效率，可以去掉显式循环，使用并行或者 RCPP 来提高 R 语言的循环速度. 若需进一步提高模拟的精确度，可增加 n_1 和 n_2 的数值，由此则必然存在某个正数，只要 n_1 和 n_2 大于该数字就可满足设定的精确性要求.

参考文献：

[1] 胡爱平，肖枝洪，苏理云，等. 抽签原理在古典概率计算中的应用[J]. 中国校外教育，2013(6)：106.

[2] 王绍光. 西方民主一个新动向：抽签的理论与实践[J]. 武汉大学学报(哲学社会科学版)，2017，70(4)：5-10.

[3] 何林蕊. 城市公共资源配置中抽签、摇号行为的法律规制研究[D]. 上海：华东师范大学，2018.

[4] 马正红. 正确认识随机抽取合理采用抽签处理[N]. 政府采购信息报，2017-09-18(011).

[5] 侯思博. 基于改进抽签法的机场时隙分配研究[D]. 南京：南京航空航天大学，2013.

[6] SULLIVAN P. The FA cup draw and pairing up probabilities[J]. The College Mathematics Journal，2016，47(4)：282-292.

[7] 程嘉炎. 乒乓球竞赛法研究[M]. 北京：人民体育出版社，1981：39-52.

[8] 王蒲，余丽华. 淘汰制抽签法研究[J]. 西安体育学院学报，2000(1)：46-50.

[9] 黄浩军，王金灿，谢雪峰，等. 乒乓球单淘汰赛"逐区双分抽签法"原理及实践研究：一种科学、合理和简便的抽签法[J]. 武汉体育学院学报，2008(10)：60-65.

[10] 魏艳华，王丙参，邢永忠. 对估计圆周率的不同蒙特卡洛模型评价与选择[J]. 统计与决策，2019，35(17)：9-13.

[11] 尤利平. 基于 k/N 模式彩票博弈模型的统计审计与随机测试[J]. 统计与决策，2014(12)：49-51.

[12] 蔡国宪，李炯天，朴光日，等. 稀疏网格配置法在随机 Burgers'方程中的应用[J]. 延边大学学报(自然科学版)，2014，40(1)：20-24.